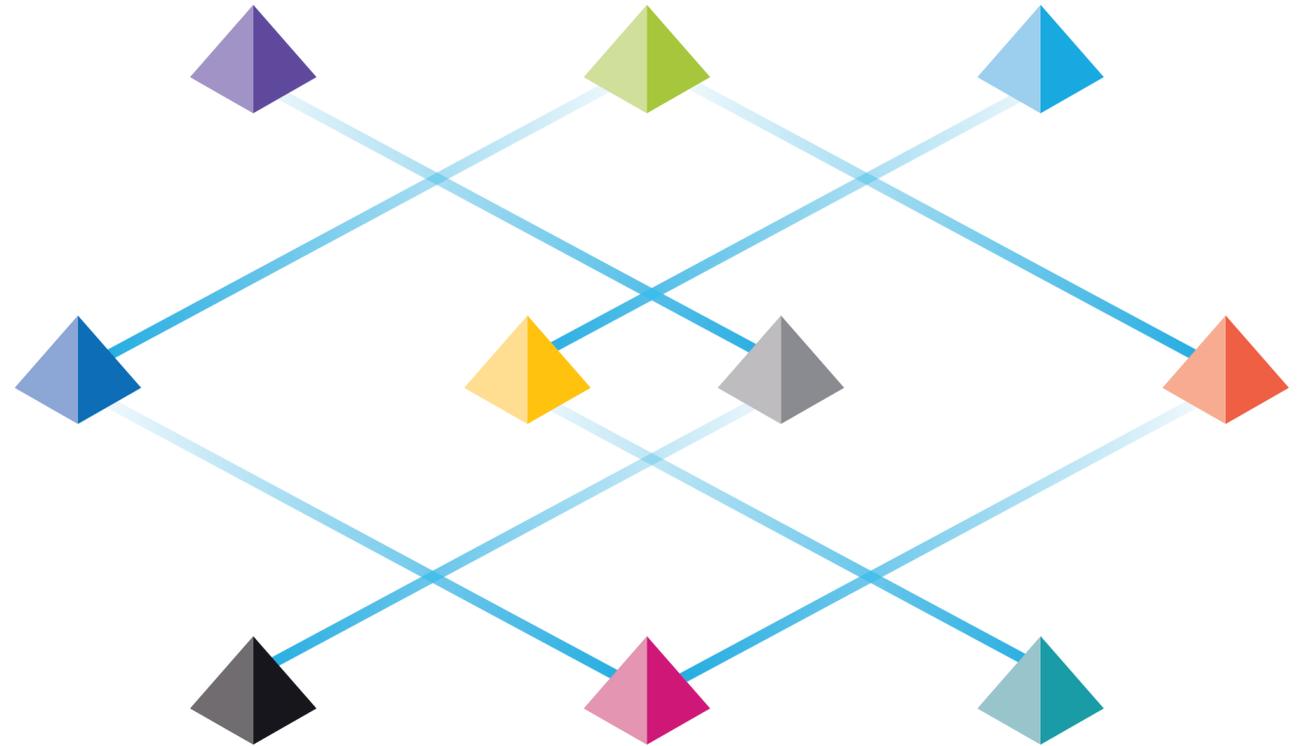
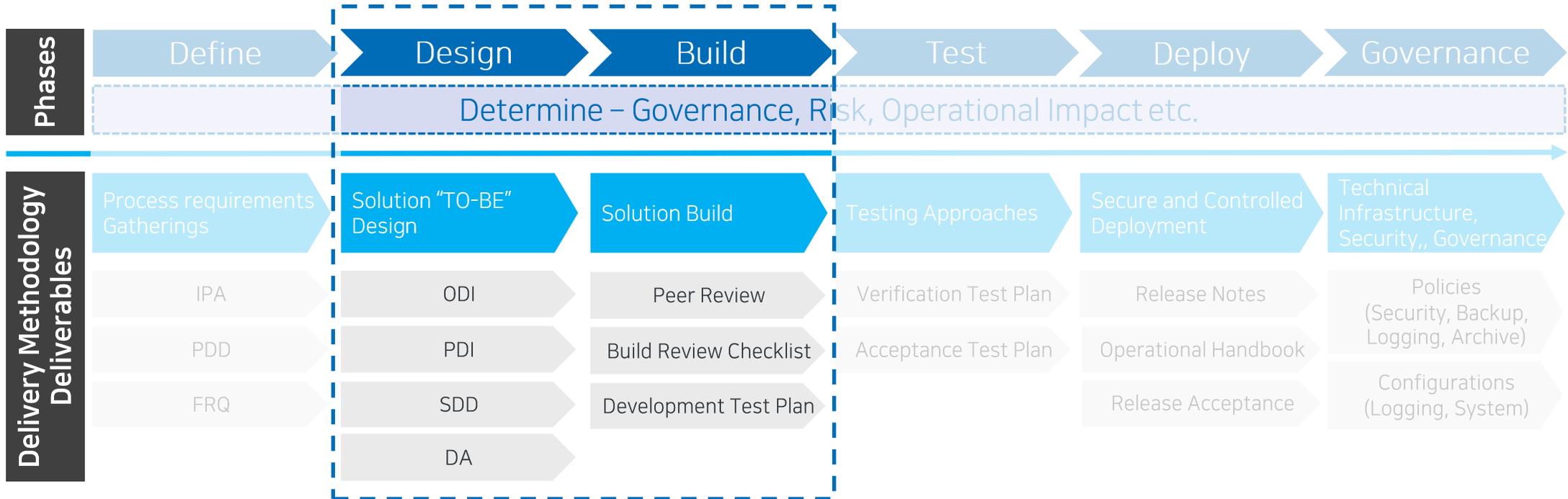


Day1. S3.
Object & Process
Solution Design



Setting up the Delivery Methodology



Contents

01

객체지향 RPA

객체지향 프로그래밍(OOP)에 대한 이해를 바탕으로 블루프리즘이 객체지향 RPA를 어떻게 실현 하는지 알아본다.

02

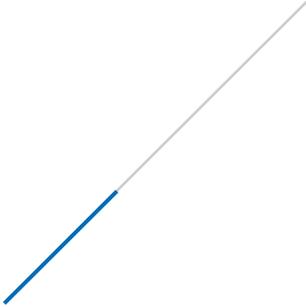
오브젝트 & 프로세스

객체지향 RPA를 실현하기 위한 핵심 개념, '비즈니스 오브젝트'에 대해 알아보고, 좋은 프로세스란 무엇인지 그리고 이를 위해 블루프리즘은 어떤 설계방법 및 기능을 제공하는지 알아본다.

03

프로세스 빌딩 / 디버깅 / 워크큐

자동화 솔루션 디자인 방법론과 기업의 자동화 프로세스를 표준화 하기 위해 블루프리즘이 제공하는 워크큐와 프로세스 템플릿을 어떻게 활용하는지에 대해 알아본다.

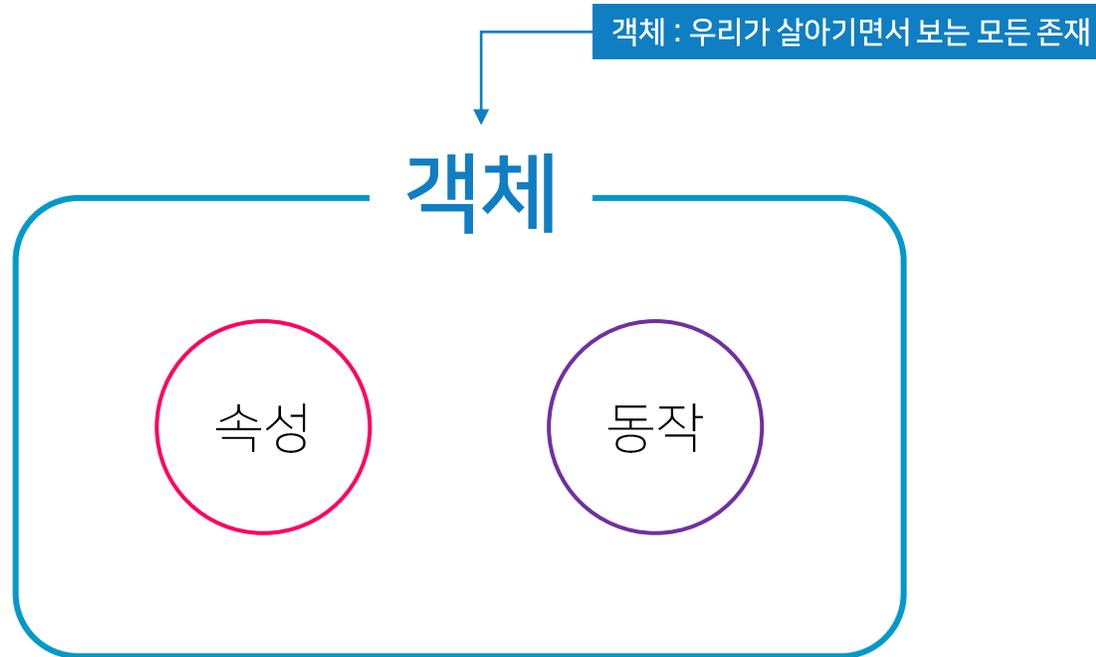


1

Blue Prism
이 추구하는
객체지향 RPA

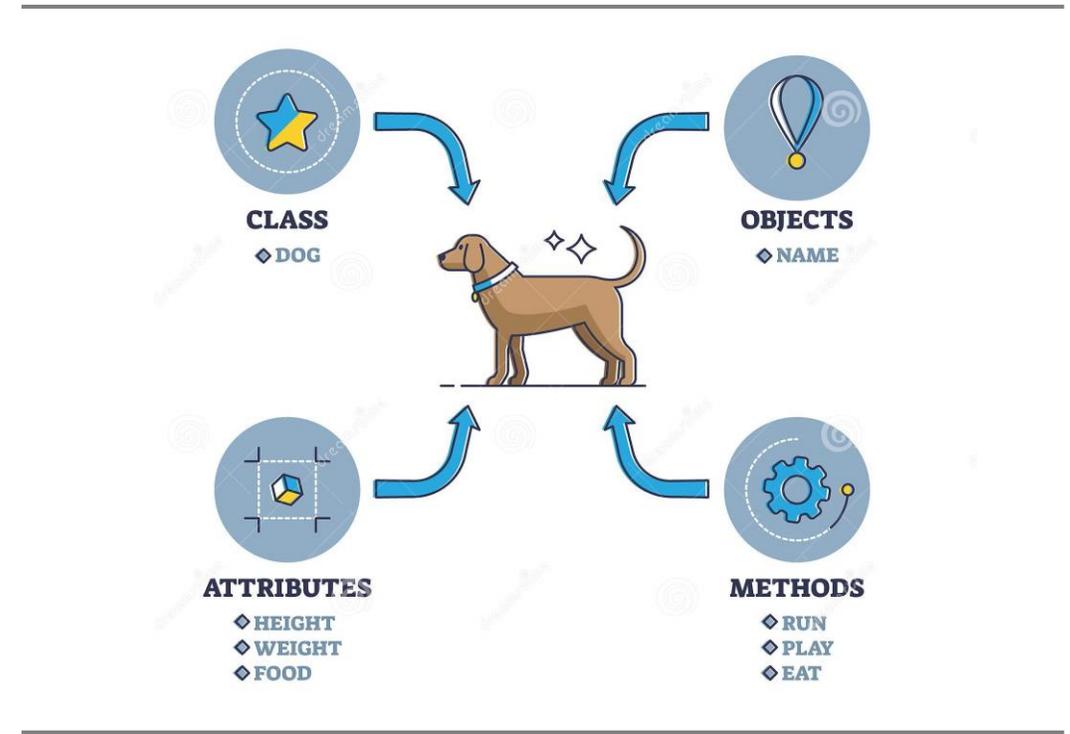
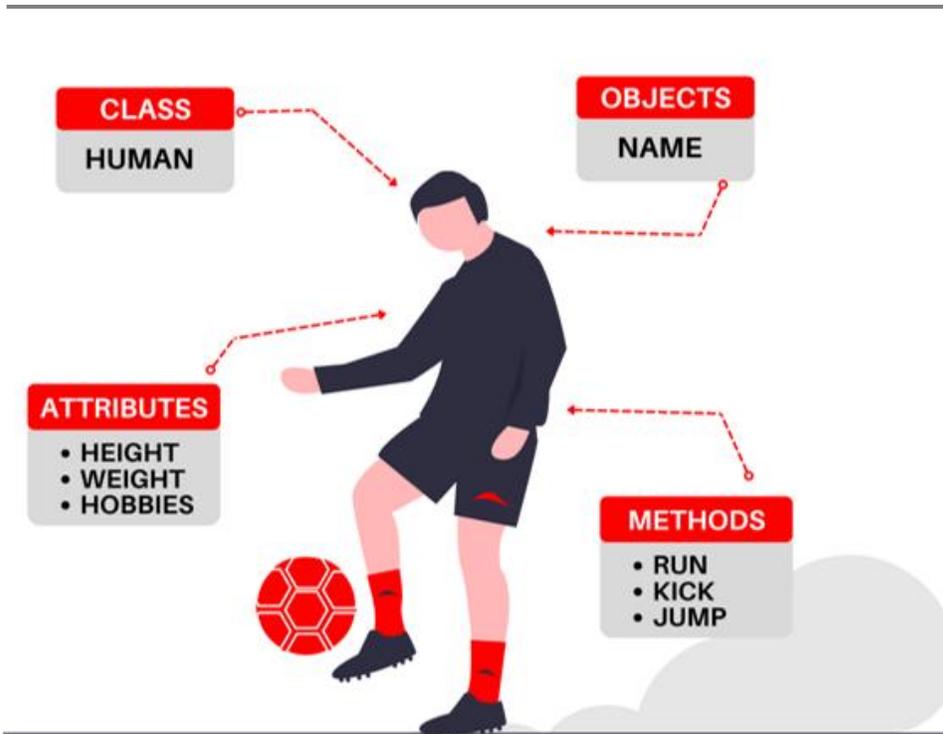
객체(Object)란?

객체는 생활하면서 주변에 있는 사람,사물 또는 추상적으로 생각할 수 있는 것들 중에서 **자신의 속성을 갖고 다른 것과 식별이 가능한 것**을 의미합니다.
객체는 **속성과 동작**으로 이루어져 있습니다.



객체(Object)의 예시

객체는 생활하면서 주변에 있는 사람, 사물 또는 추상적으로 생각할 수 있는 것들 중에서 **자신의 속성을 갖고 다른 것과 식별이 가능한 것**을 의미합니다.
객체는 **속성**과 **동작**으로 이루어져 있습니다.



객체지향 프로그래밍이란?

프로그램을 여러 개의 독립된 객체들과
그 객체들 간의 상호작용으로 파악하는
프로그래밍 접근법



프로그램을
객체들과 객체들 간의 소통으로
바라보는 것

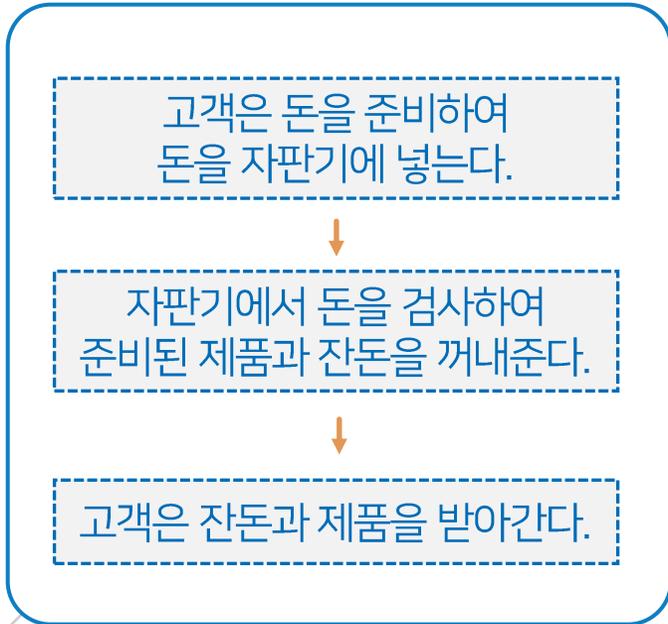


객체지향 프로그래밍을 하기 위해서는?

1. 프로그램에 어떤 객체들이 필요할지 정한다
2. 객체들의 속성과 동작을 정한다
3. 객체들이 서로 어떻게 소통할지 정한다

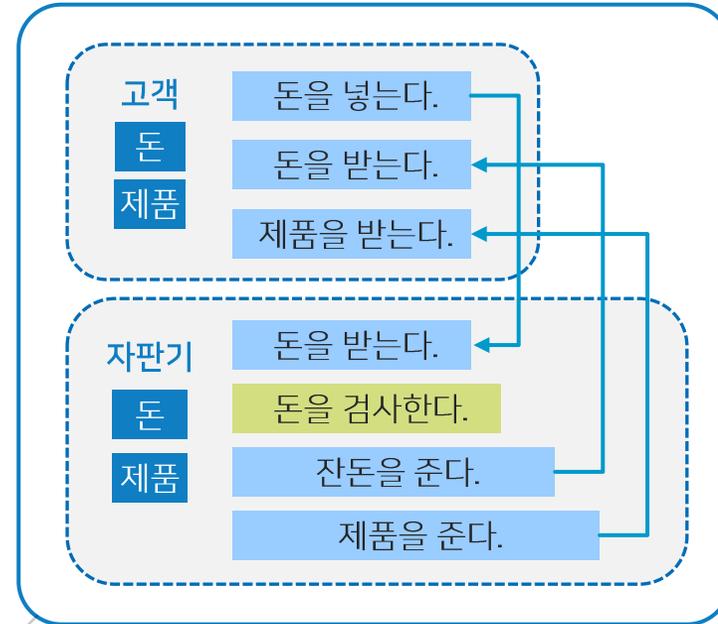


절차지향 & 객체 지향 프로그래밍 비교



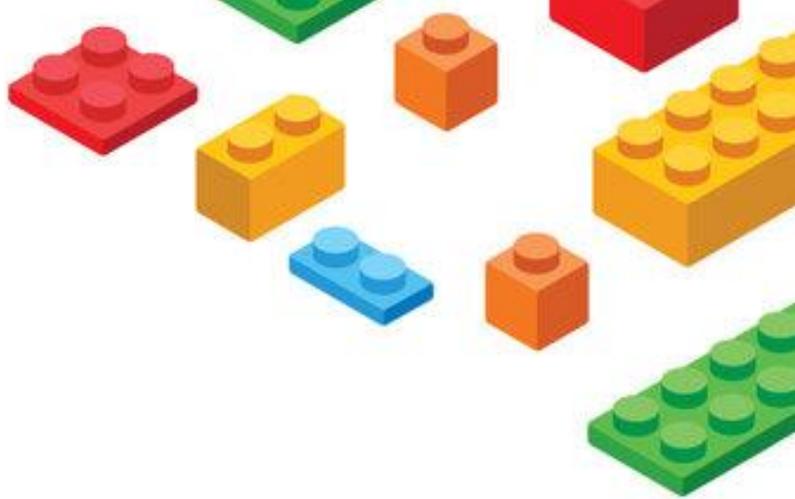
절차지향 분석

데이터를 중심으로 순차적 실행에 초점을 맞추고 코드를 작성.
 돈을 지불하는 절차, 계산하는 절차, 제품을 받는 절차를 중점으로 함수가 구성됨



객체지향 분석

프로세스의 주요 행위자인 고객과 자판기를 객체로 설정하고 돈과 제품에 대한 관계에 따라 객체를 중심으로 함수를 작성



객체지향 프로그래밍의 이점



재사용성

상속을 통해서 프로그래밍 시 코드의 재사용을 높일 수 있습니다.



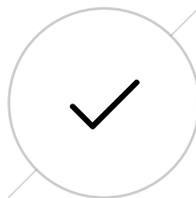
생산성 향상

잘 설계된 클래스를 만들어서 독립적인 객체를 사용함으로써 개발의 생산성을 향상시킵니다. ctrl + v 실행은 나중에 수정사항이 생겼을 때 일일이 찾아서 바꿔주어야 하지만, 클래스는 한 곳에서만 바꾸면 됩니다.



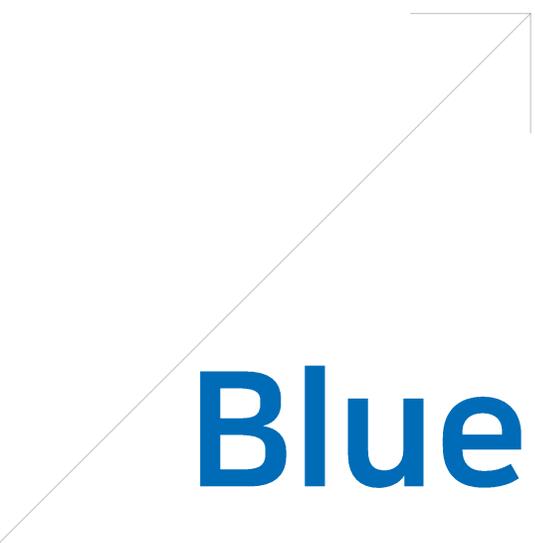
자연적인 모델링

실제 세상의 사물들을 하나의 객체로 생각하며 프로그래밍이 이루어지기 때문에 클래스를 구현할 때 일상생활의 모습과 대비시키며 자연스럽게 구현할 수 있습니다.



유지보수의 우수성

캡슐화로 인해 모듈끼리의 의존성이 떨어집니다. 그렇기에 프로그램 추가 및 수정과 같은 유지보수에 용이합니다.



Blueprism

RPA



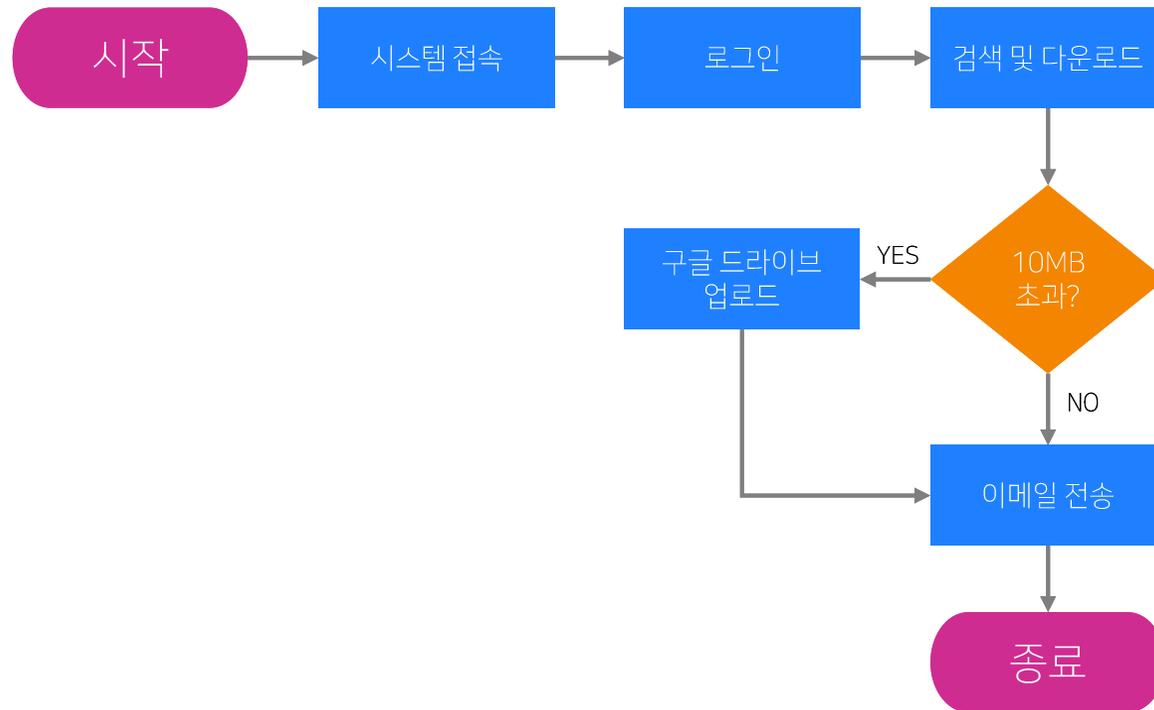
블루프리즘의 객체지향 RPA란?

블루프리즘은 객체지향 RPA를 위해
오브젝트와 프로세스를 분리하고,
오브젝트는 객체지향 프로그래밍의
방식을 따르도록 설계, **프로세스**는
작업의 흐름만을 담당하도록 설계.

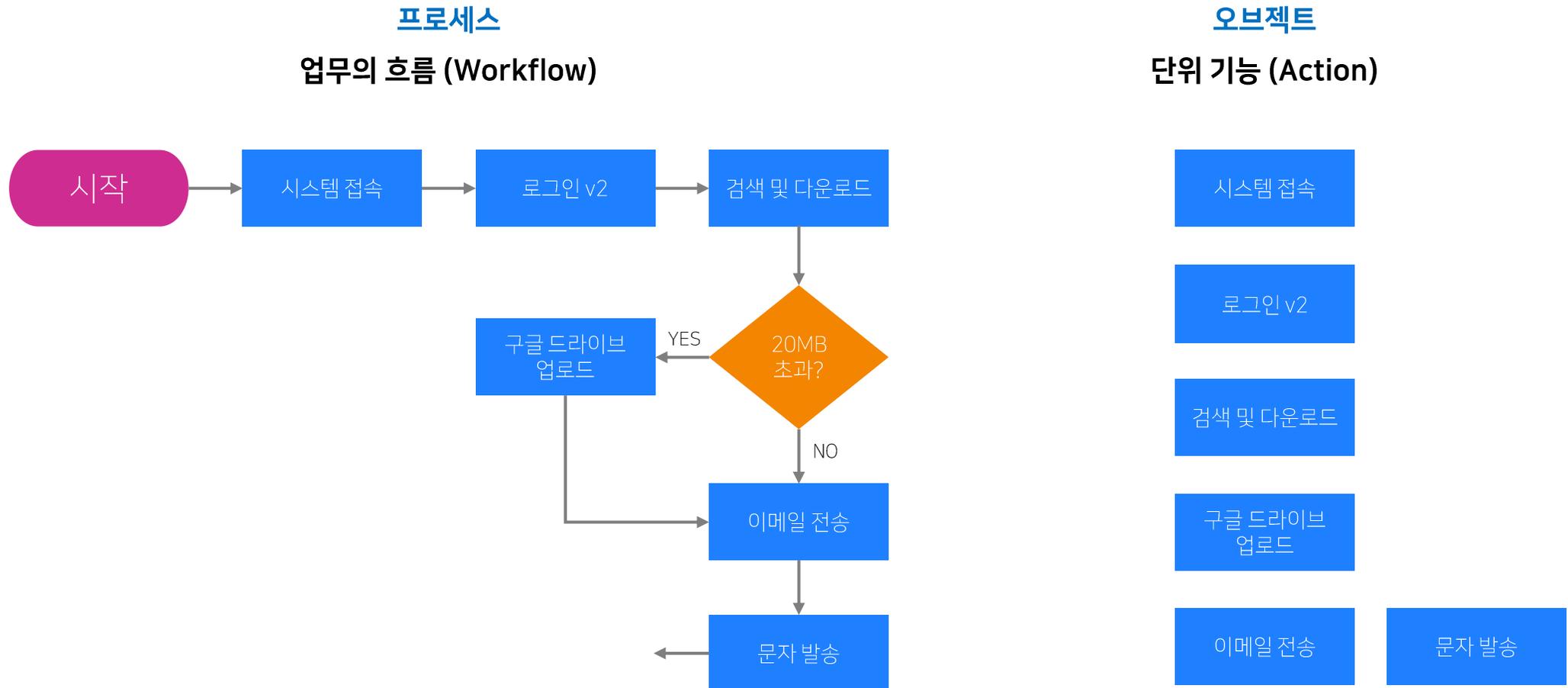
자동화 프로세스의 구성 - 개념 이해하기

프로세스
업무의 흐름 (Workflow)

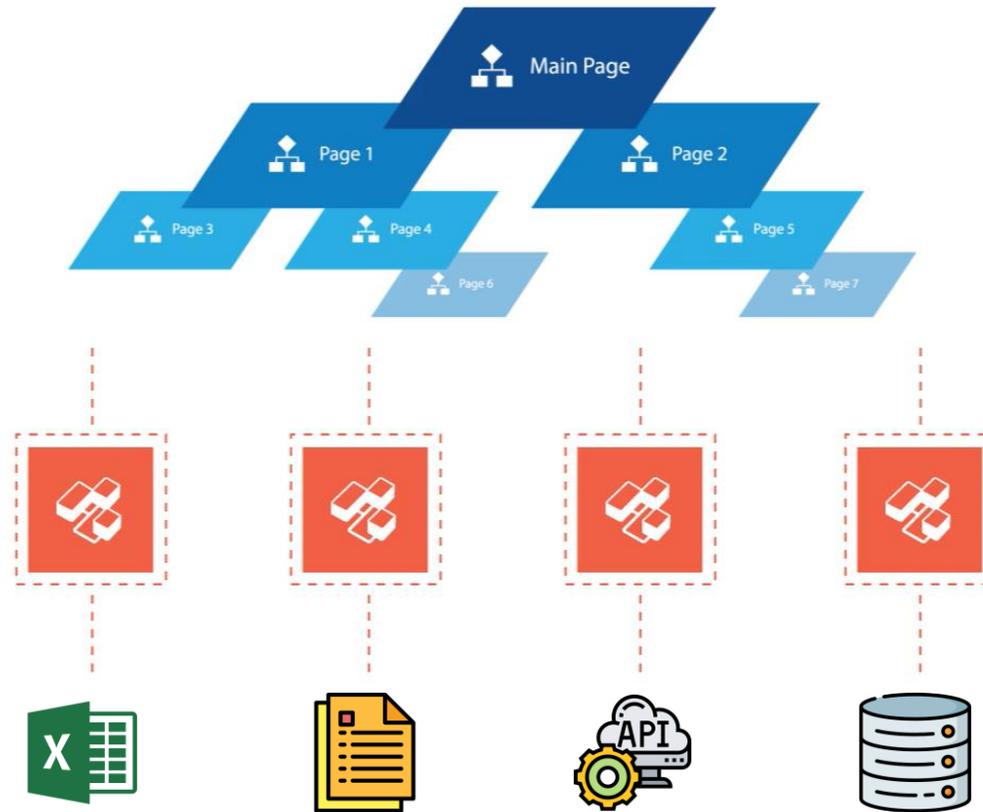
오브젝트
단위 기능 (Action)



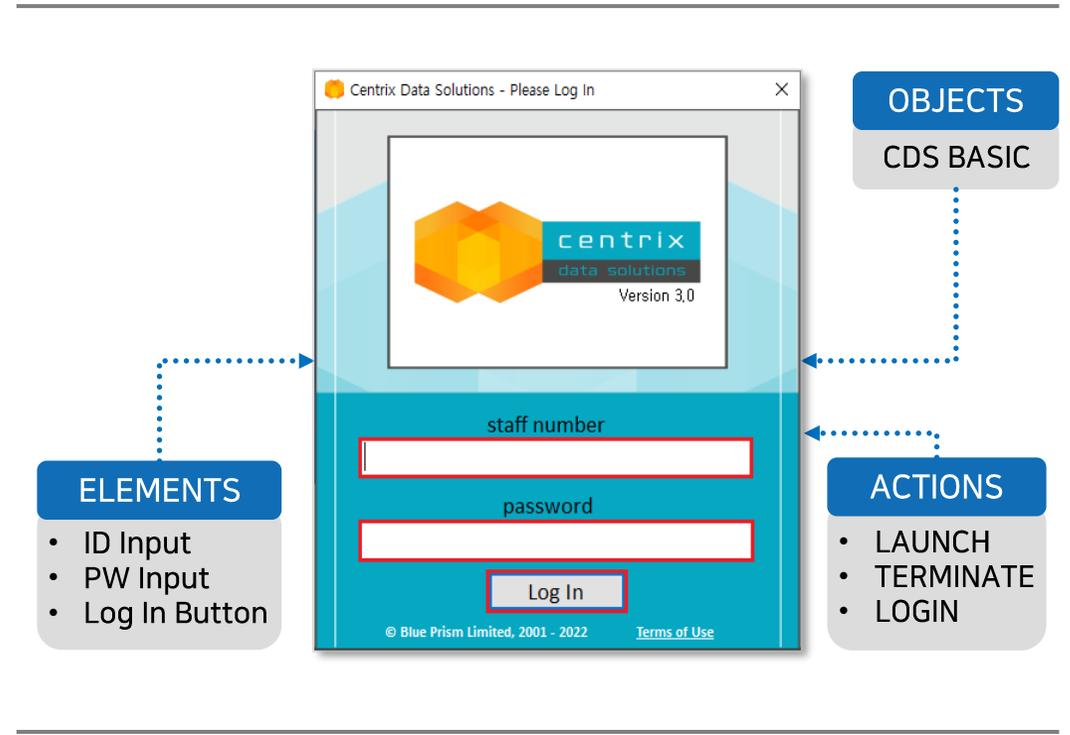
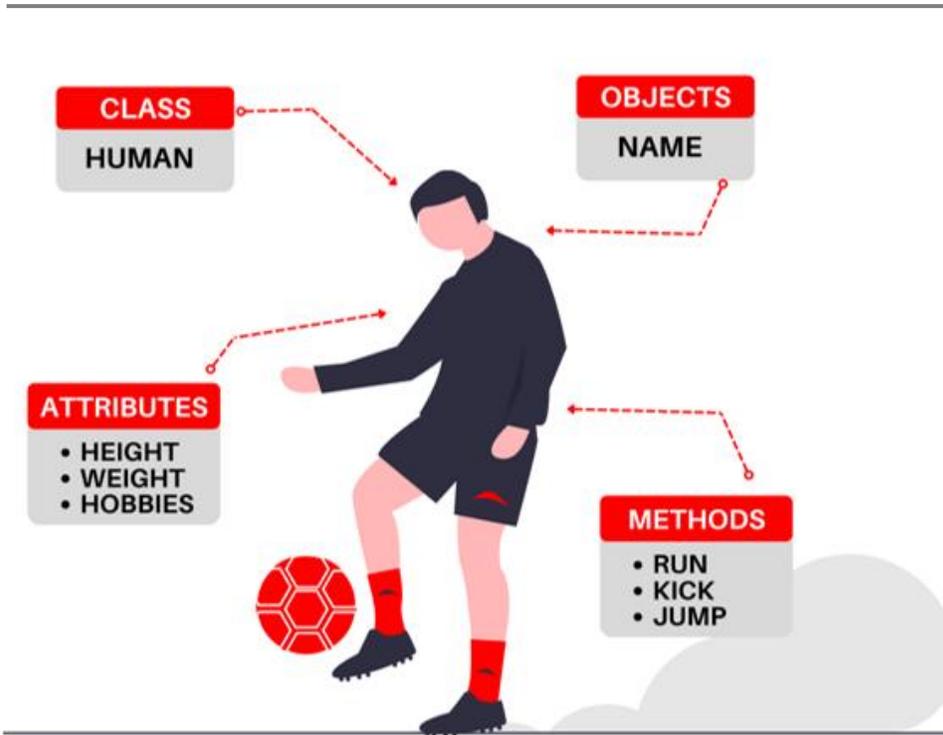
자동화 프로세스의 구성 - 개념 이해하기



블루프리즘 프로세스의 구성

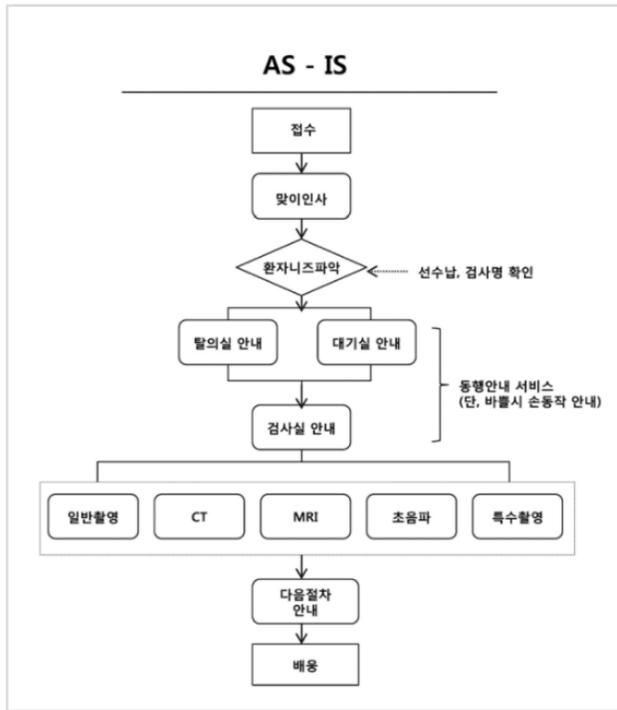


객체지향 비즈니스 오브젝트 이해

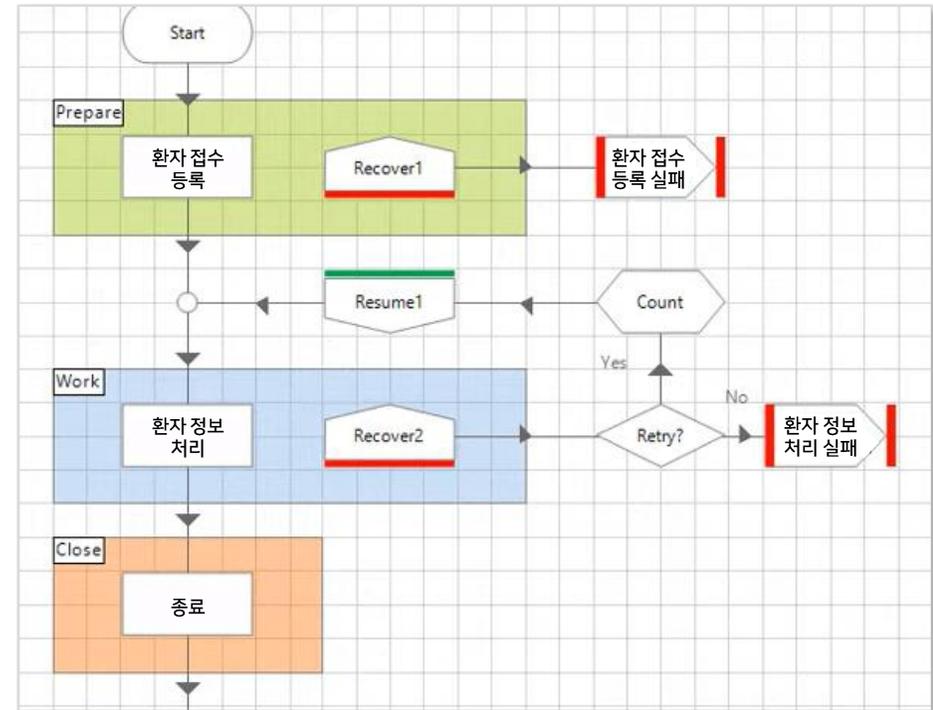


블루프리즘 프로세스

예시) 영상의학과 업무 As-Is 프로세스

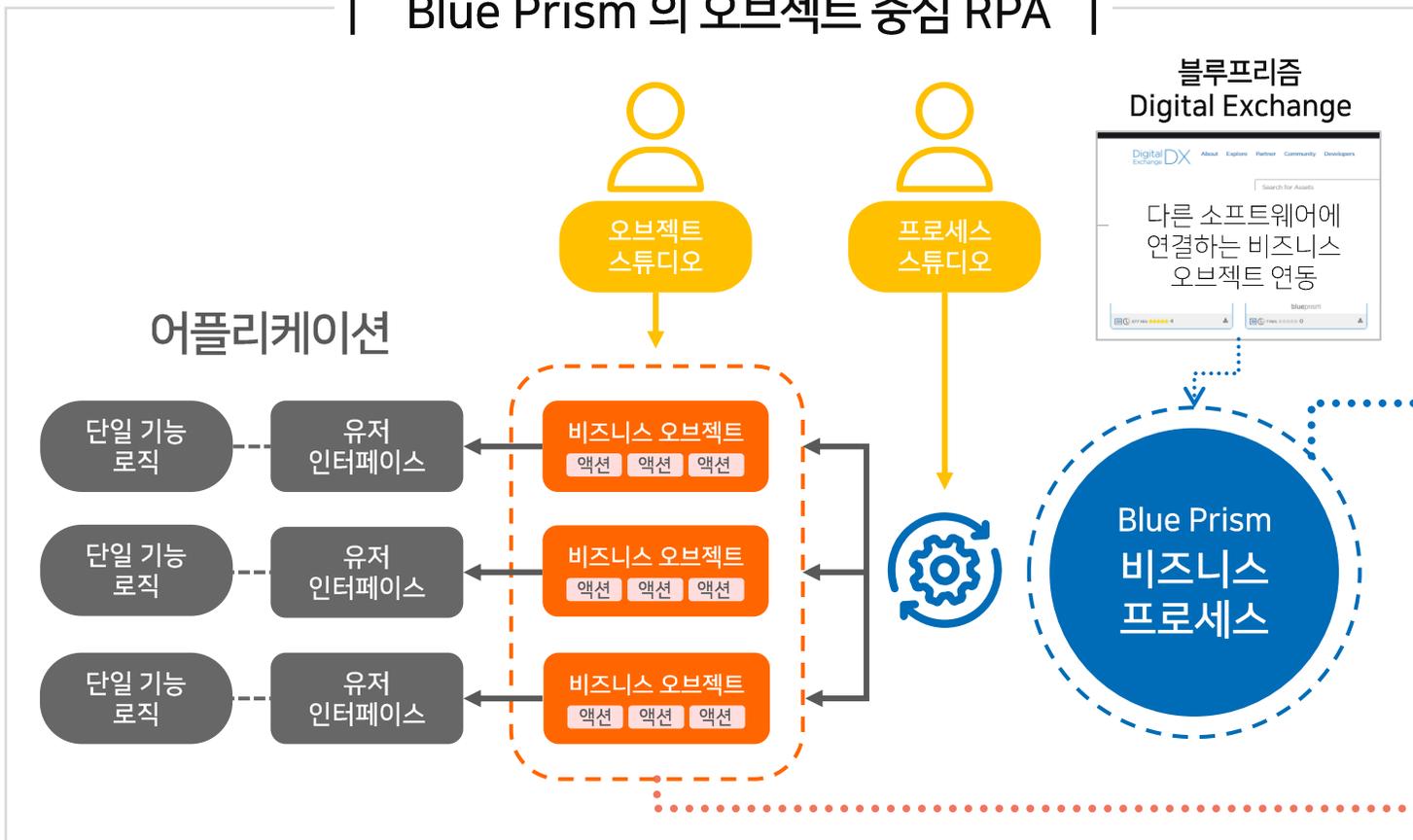


프로세스의 모습 in Process Studio

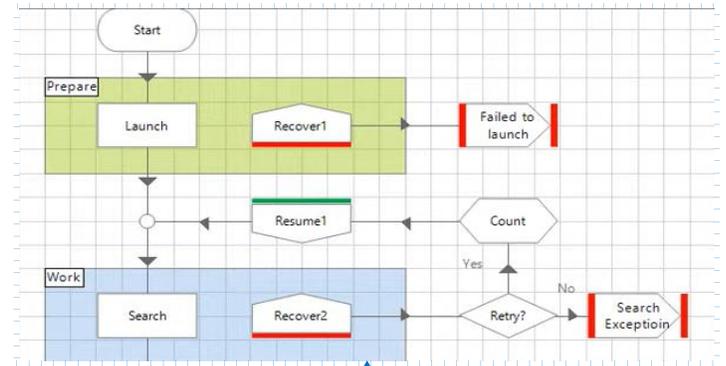


블루프리즘 프로세스의 구성

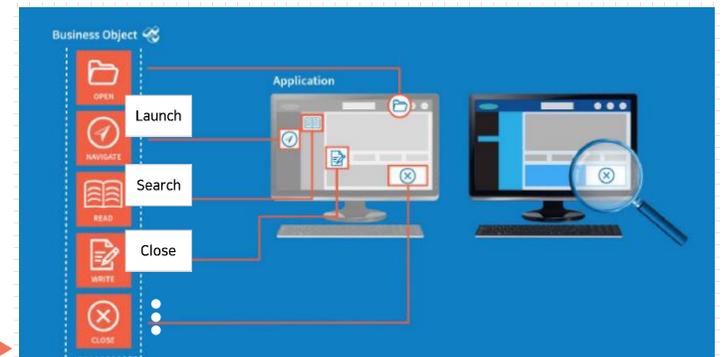
Blue Prism 의 오브젝트 중심 RPA |



블루프리즘 프로세스 | Blue Prism Process



비즈니스 오브젝트 | Business Object



블루프리즘 프로세스의 구성



Business Object



Business Objects



Process



2-1

블루프리즘의
비즈니스 오브젝트

What's

Blueprism Business Objects



비즈니스 오브젝트란?

비즈니스 오브젝트는 프로세스 흐름도에서 웹 서비스와 함께 외부 시스템과 상호 작용하는 수단을 제공한다.

비즈니스 오브젝트는 프로세스에서 호출할 수 있는 기능을 Action으로서 제공한다. 예를 들어, 사내 시스템에 로그인 하는 기능을 수행하는 Action이 있다면 프로세스에서는 로그인 하고자 하는 계정정보를 매개변수로 넘겨주며 해당 Action을 호출하면, 대상 시스템에 로그인 하는 기능을 수행할 수 있다.

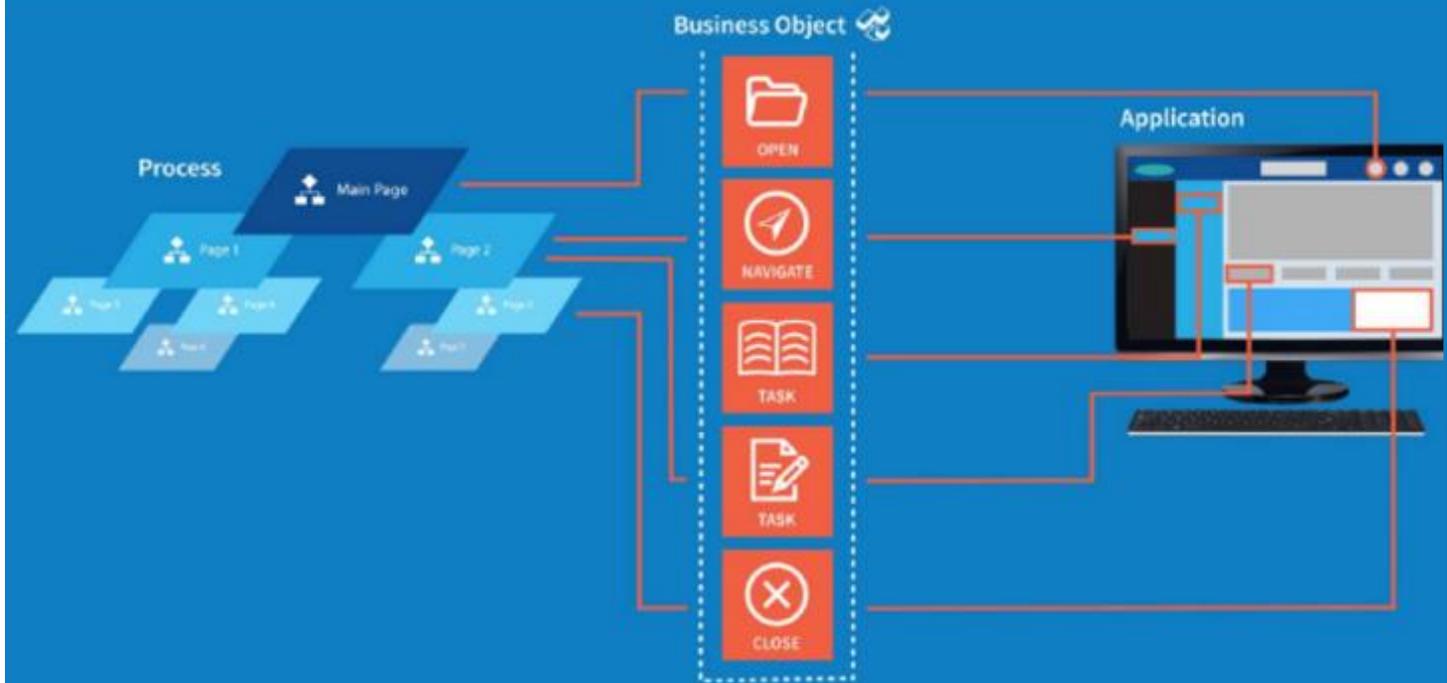
이러한 방식으로 비즈니스 오브젝트에 Action을 통해 기능을 제공하면 동일한 프로세스에서 여러 번 사용 되거나 다른 프로세스들에서도 여러 번 사용 될 수 있다. 이는 객체지향 RPA의 기본 원리이며, 이를 통해 재사용성을 높일 수 있다.

블루프리즘의 비즈니스 오브젝트 구현은 프로세스 구현과 분리된다. 이는 대상 시스템을 새 시스템으로 교체하는 경우 비즈니스 오브젝트의 재구현이 필요하지만 프로세스의 변경은 필요 없음을 의미한다.

이러한 이점을 최대한 활용하려면 구현에 있어 비즈니스 로직을 분리해야 한다. 대상 시스템들과 상호작용 하기위해 비즈니스 오브젝트를 사용하고, 데이터 흐름 및 큐 아이템 처리 등을 위해 프로세스를 사용한다.

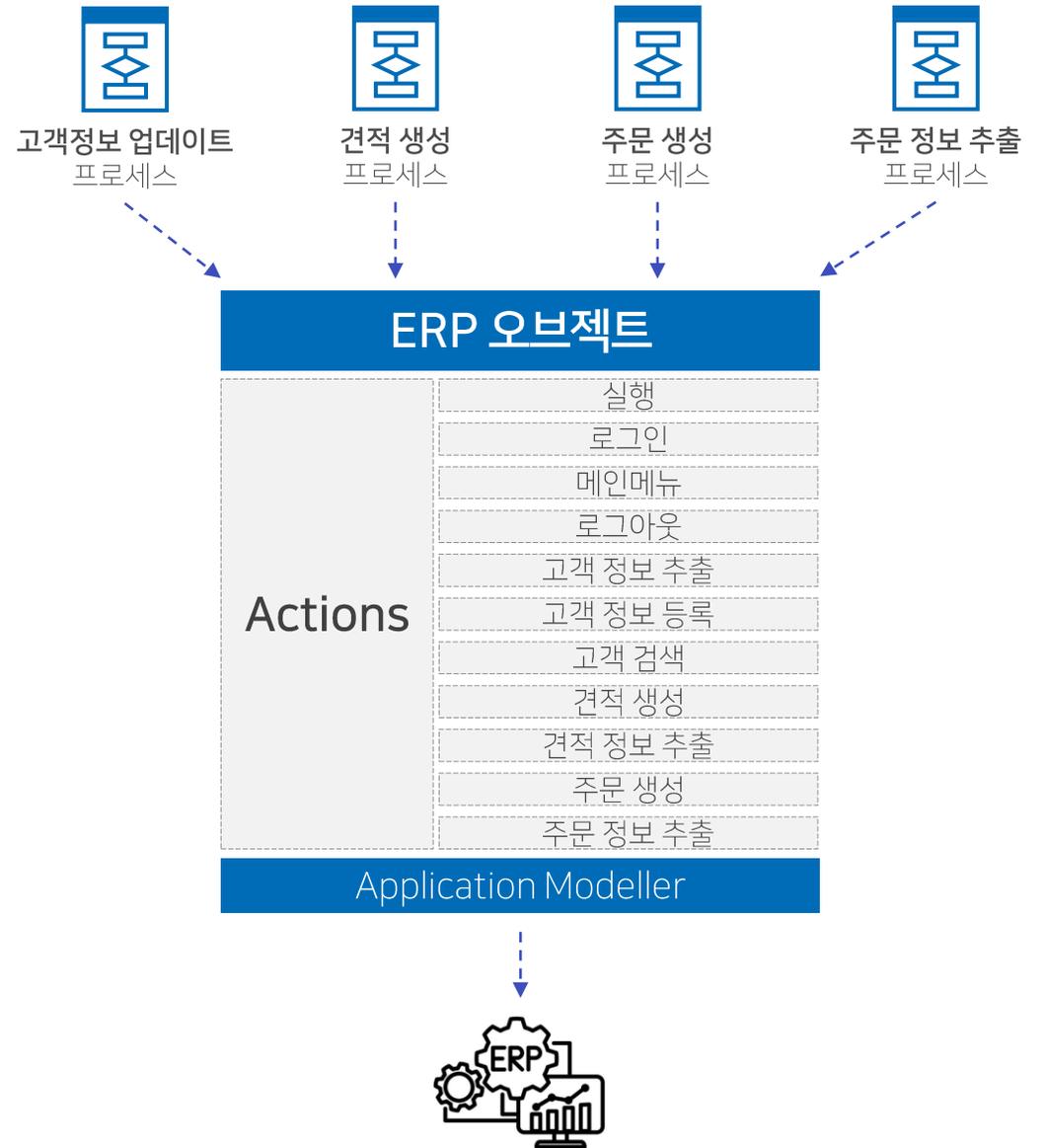
오브젝트의 특징

- | 어플리케이션 모델러를 통해 대상 어플리케이션을 모델링
- | 단독으로 사용되지 않고 프로세스의 일부로 사용 됨
- | Action이라 불리는 계층이 없는 단순한 구조로 단일 기능만을 제공
- | 각 Action은 대상 어플리케이션과 상호작용. 프로세스는 이를 호출하여 상호작용



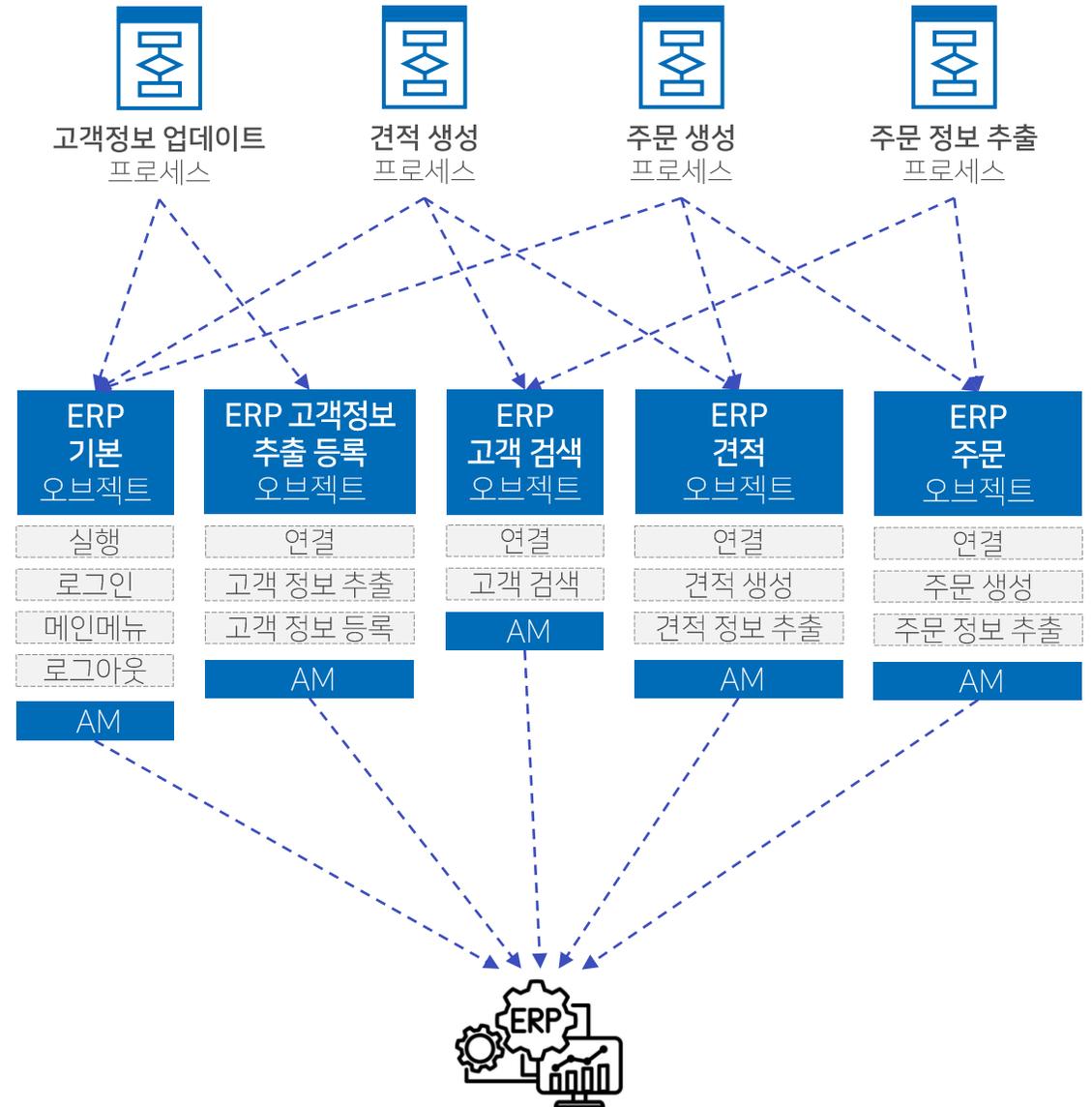
단일 오브젝트 디자인

- | 한 번에 한 명의 개발자만 오브젝트를 작업할 수 있음
- | 불필요한 작업으로 인한 메모리 낭비
- | 오브젝트의 수정은 곧 모든 프로세스에 대한 변경
- | 필요 이상으로 큰 오브젝트로 인한 DB 비효율성 증가
- | 프로덕션으로 이관되지 않을, 테스트 및 소규모 PoC 등을 위한 프로젝트에 적합



다중 오브젝트 디자인

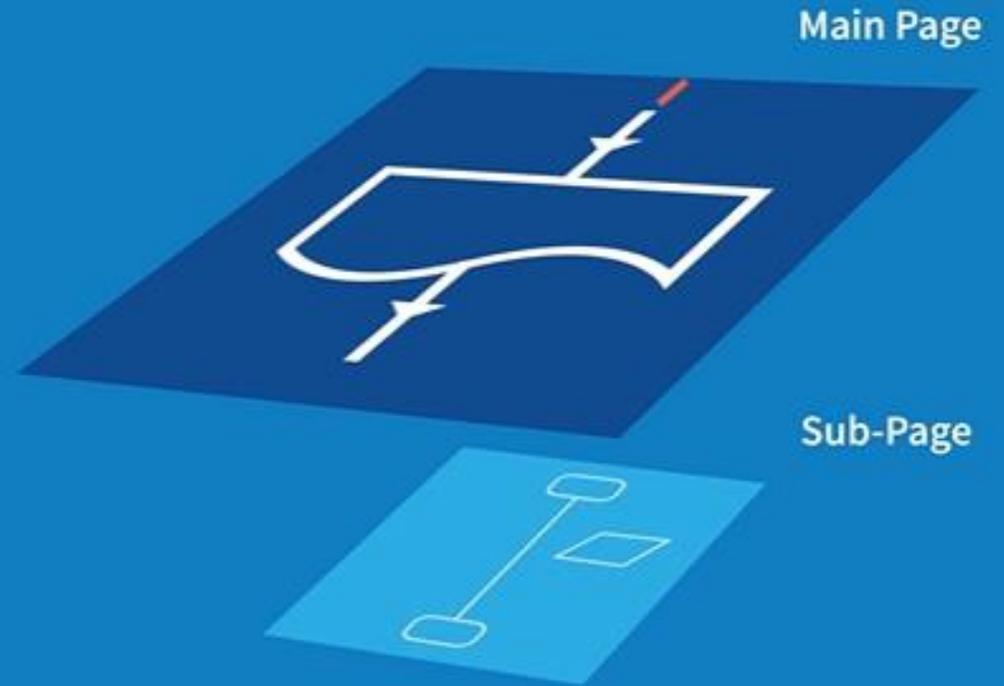
- | 한 번에 여러 명의 개발자가 ERP 오브젝트를 작업
- | 메모리 효율성 증가
- | 오브젝트의 수정은 적은 프로세스에 영향
- | 효율적인 데이터베이스 사용
- | 프로덕션에 배포 될 프로세스나 여러 명의 개발자가 참여하는 PoC에 적합.



입력 매개변수 & 출력 매개변수

- | 프로세스의 Action Stage, 비즈니스 오브젝트의 Action Page 간의 제어
- | 비즈니스 오브젝트는 가능한 일반적으로 구현되어야 함
- | 비즈니스 오브젝트는 어떠한 비즈니스 데이터도 저장해서는 안 됨
- | 비즈니스 오브젝트에서 필요한 비즈니스 데이터는 입력 매개변수를 통해 전달받고, 필요 시 출력 매개변수로 데이터를 전달

* 위와 같이 구현 함으로서 오브젝트의 **재사용성이 보장** 됨.

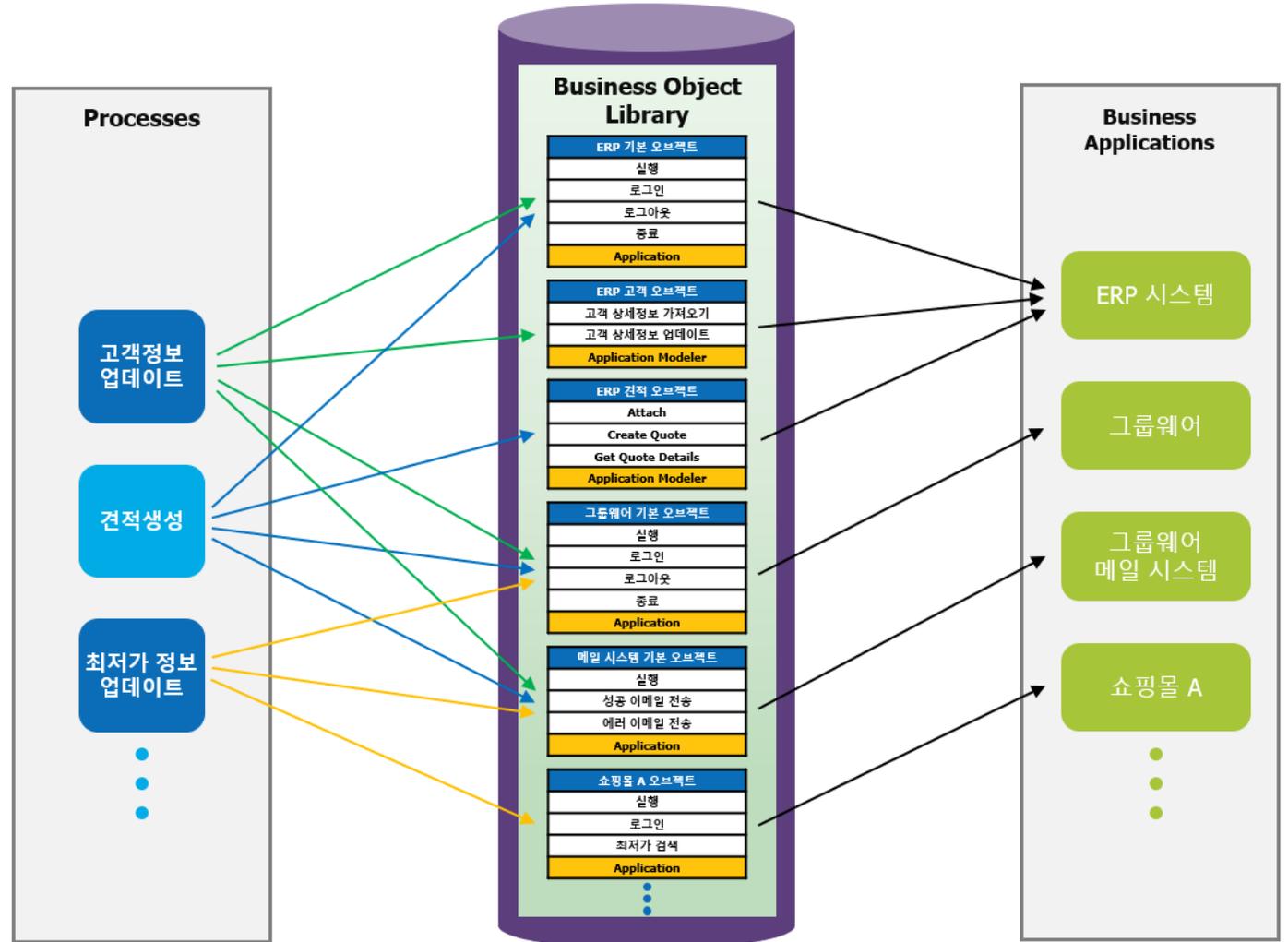


Business Object Library (BOL)

조직의 RPA가 확산됨에 따라 오브젝트는 많아지게 되고, 누적된 재사용성 높은 오브젝트들은 RPA의 확산 속도를 가속화하기도 하지만, 많은 오브젝트를 잘 관리하는 것 또한 중요하다.

Business Object Library(BOL)는 기개발된 오브젝트들을 한 눈에 볼 수 있는 툴로, 설계 단계에서 기개발된 오브젝트를 재사용 할지, 추가 Action을 개발할지, 새로운 오브젝트를 생성할지 등에 대한 결정을 할 수 있다.

개발 및 운영 단계에서 오브젝트에 변동이 생겼을 경우 즉시 업데이트를 해주어 최신성을 유지해 주는 것 또한 중요하다.



BOL Template

BUSINESS OBJECT LIBRARY

Application Type (Technology)	Business Object Name	Actions	Inputs	Outputs	
Application Type 1	APP 1 VBO 1	Launch			
		Attach		Attached	
		Exit			
	APP 1 VBO 2	Attach			Attached
		Login		Username Password	
		Log out			
Get Logged in Status				Logged In Status	
Application Type 2	APP 2 VBO 1	Attach		Attached	
		Select Menu			
		Select Option	Option Name		
Application Type 3	APP 3 VBO 1	Launch			
		Attach		Attached	
		Exit			
	APP 3 VBO 2	Attach			Attached
		Login		Username Password	
		Log out			
Application Type 4	APP 4 VBO 1	Get Logged in Status		Logged In Status	
		Attach		Attached	
		Select Menu			
		Select Option	Option Name		



Business
Library

Business Object
Library

Business Object
Library

Business Object
Library



비즈니스 오브젝트 Best Practice

5가지 황금 룰



다중 오브젝트 디자인

동시에 여러 개발자가 오브젝트 개발 가능, 메모리 및 데이터베이스 효율성 증가, 오브젝트의 수정은 적은 프로세스에 영향 등의 이점.



단일 기능으로 제한

단일 기능만 수행하도록 개발된 Action은 보다 직관적이며 수정 시 최소한의 영향도를 가져, 유지보수에 용이.



비즈니스 로직 제거

특정 업무에 국한된 비즈니스 데이터 또는 로직을 제거함으로써 오브젝트는 일반적인 설계가 되며, 결과적으로 재사용 성이 높아짐.



입력 매개변수 사용

Action 수행에 필요한 비즈니스 데이터 또는 로직을 입력 매개변수를 통해 Action에 전달하도록 설계.



출력 매개변수 사용

Action 수행의 결과는 출력 매개변수를 통해 호출자에게 반환. 프로세스는 해당 값에 따라 업무의 흐름을 결정.



블루프리즘 비즈니스 오브젝트



재사용성

어떤 자동화 프로세스 에도 구매 받지 않는, 대상 어플리케이션 또는 시스템과 상호작용하는 독립적인 오브젝트를 생성함에 따라 비즈니스 로직과 분리되어 높은 재사용성을 보장함.



확장성

재사용성이 높은 비즈니스 오브젝트들이 축적됨에 따라 새로운 자동화 프로세스를 제작하기 용이하며, 궁극적으로, 사내 코어 시스템들이 모두 오브젝트화 되었을 때 프로세스 정의만으로 손쉽게 확장할 수 있음.



효율성

객체지향 개념에 충실한 비즈니스 오브젝트 설계로 대상 어플리케이션의 변화에 대응하기 위해 프로세스를 수정할 필요 없이 오브젝트의 수정만으로 효율적으로 대응 가능.



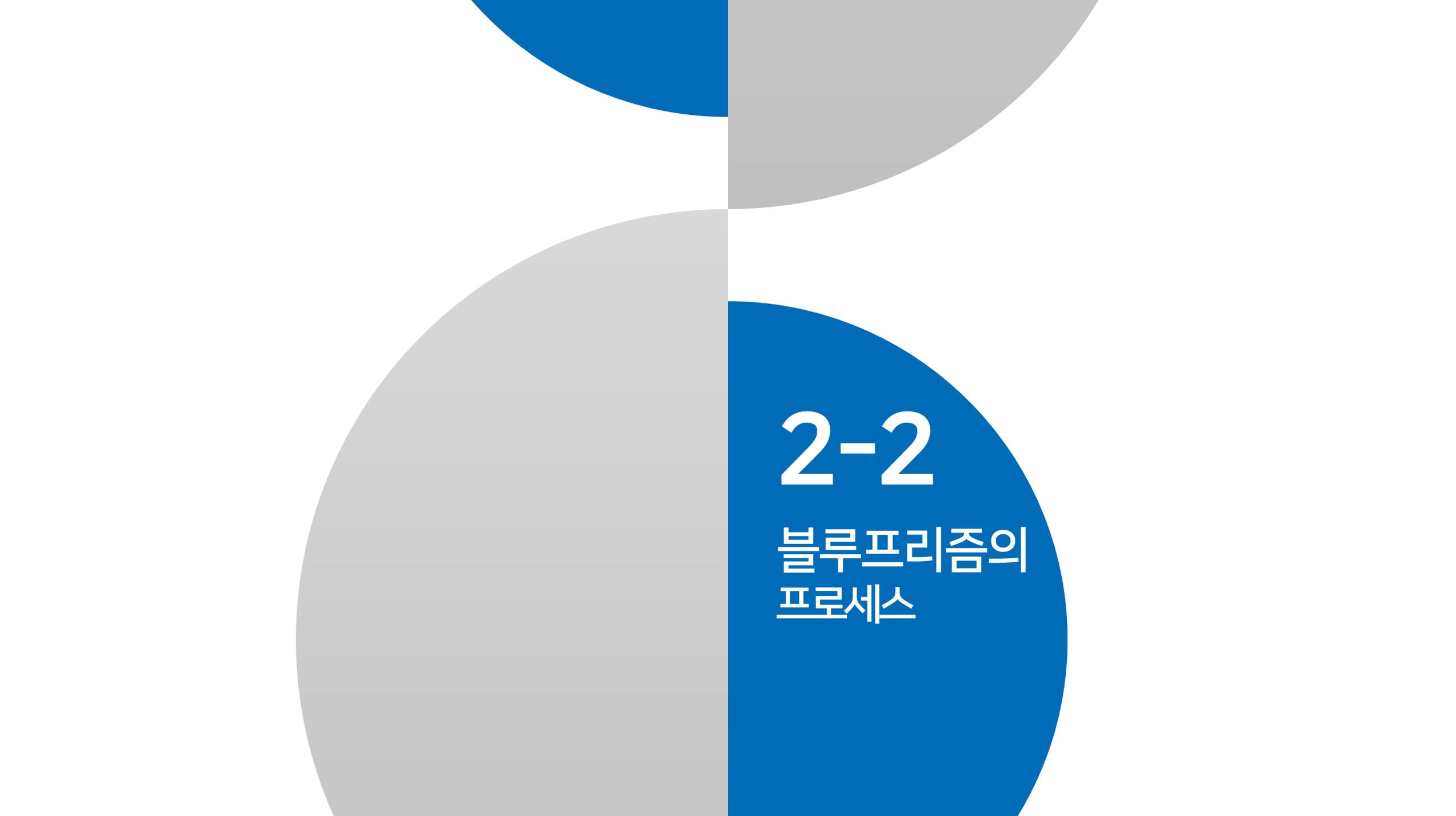
유지보수 용이성

단일, 단순한 기능으로 제한된 Action들은 디버깅/수정 시 고려해야 할 Side-effects의 예상이 쉽고 직관적이기 때문에, 개발 공수가 줄어들고 안정성 또한 쉽게 확보할 수 있음.



표준화

기업의 거버넌스를 따르며 일관성 있는 설계로 구현된 오브젝트들은 기업의 자동화 시스템을 안정적이고 지속적으로 운영 및 확신을 할 수 있도록 표준화를 하는데 큰 기여를 함.



2-2

블루프리즘의
프로세스

What's

Blueprism Processes



블루프리즘 프로세스란?

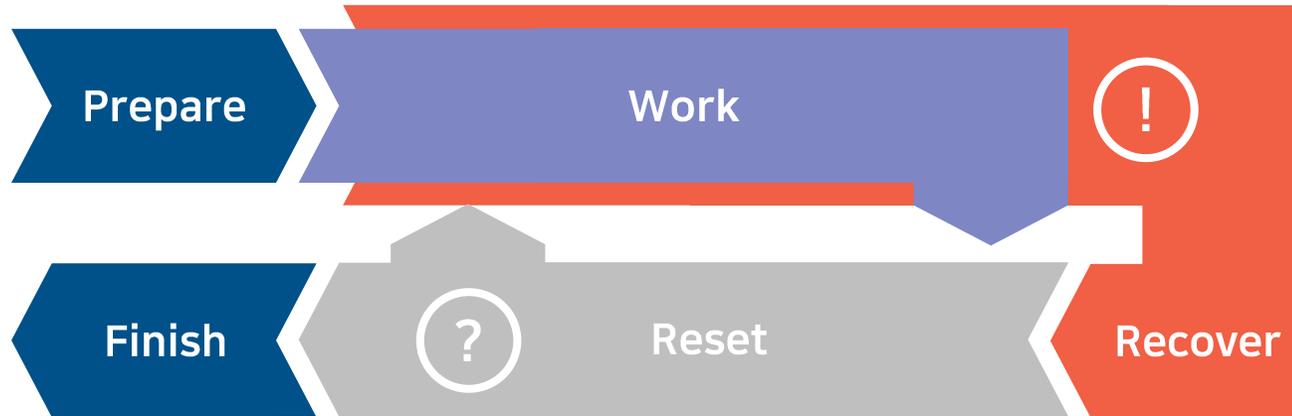
블루프리즘의 자동화 프로세스는 업무의 흐름을 쉽게 파악할 수 있는 워크플로우 기반으로 설계된다.

프로세스는 대상 어플리케이션과 상호작용을 담당하는 비즈니스 오브젝트에 정의된 Actions를 호출하여 업무의 흐름을 정의한다. 비즈니스 또는 시스템 예외가 발생했을 경우 어떻게 처리하는지에 대한 정의도 필요하다.

프로세스의 핵심은 예외 발생 시 이를 처리할 수 있어야 하고 예상치 못한 예외가 발생하더라도 강제종료가 되는 않으며 탄력적으로 복귀하여 기존의 작업을 이어갈 수 있도록 설계해야 한다.

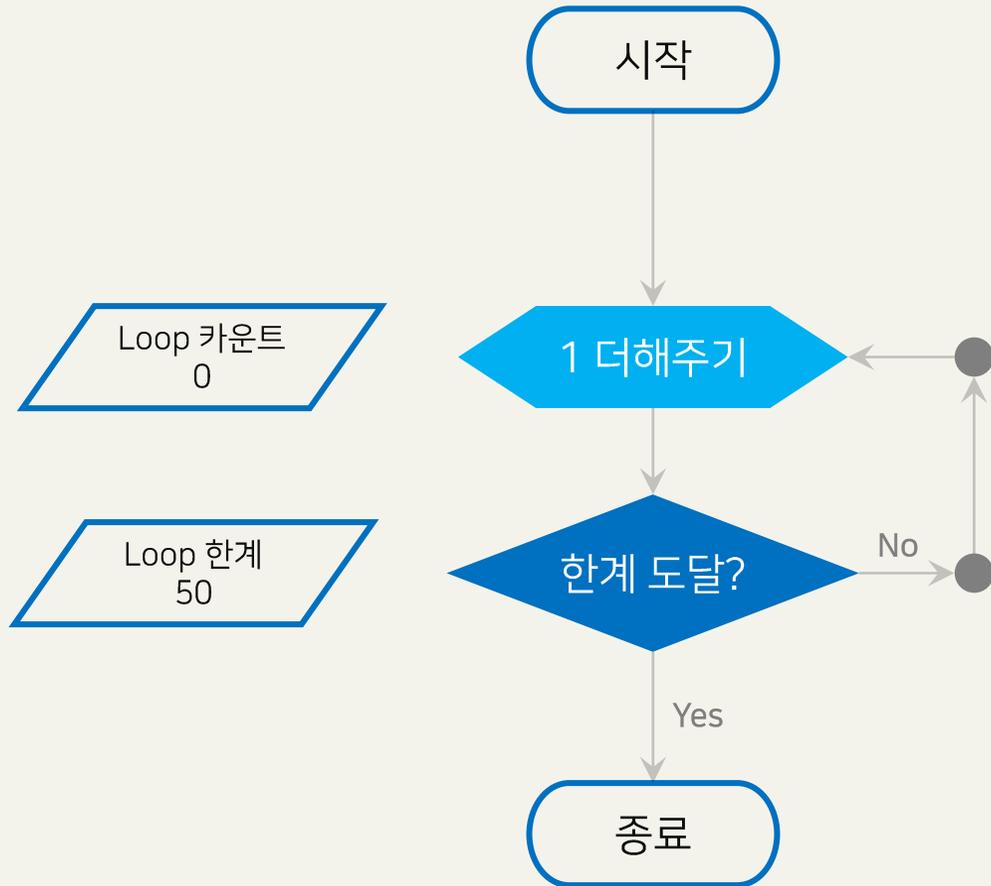
블루프리즘은 신뢰성 높고 탄력적인 프로세스를 구성할 수 있는 프로세스를 만들기 위한 '예외처리 기법', '프로세스 템플릿', '워크 큐' 등과 같은 기능들을 제공한다.

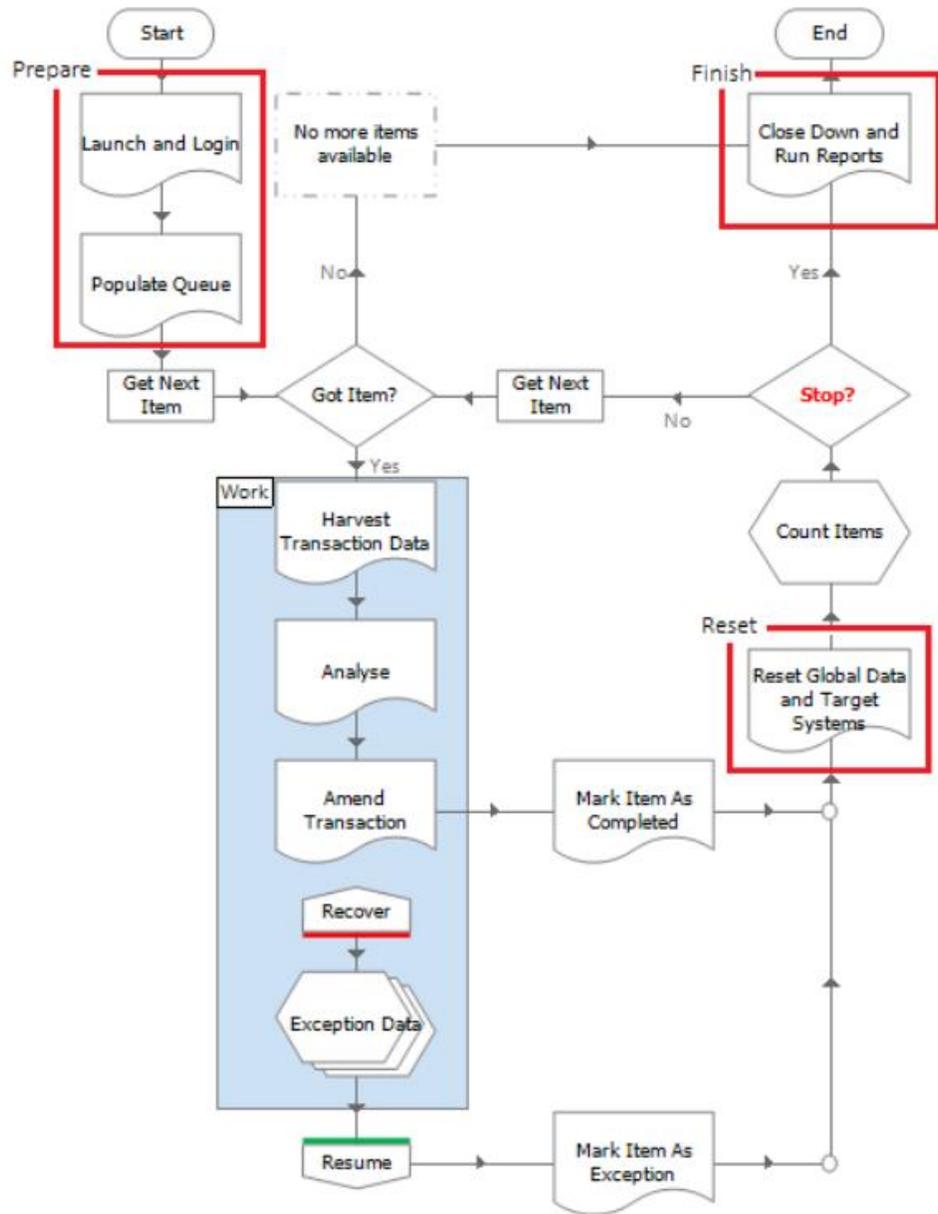
블루프리즘 프로세스의 컨셉



순환 경로 Circular Path

다량의 트랜잭션을 처리하는 프로세스의 기본은 반복 작업.
블루프리즘의 프로세스의 골격이 어떻게 구체화 되는지
알아보기 전에 가장 기본적인 순환 경로의 이미지를 기억한다.





프로세스 템플릿

Process Template

새 프로세스를 시작하기 위한 기본 프레임 워크.

- 프로세스 개발 시간 단축
- 모든 프로세스에 일관성을 제공하여 업무의 흐름 파악 및 유지보수 용이
- 개발자가 쉽게 블루프리즘 프로세스의 Best-practice와 조직의 개발 방법론을 준수

[Download Template](#)

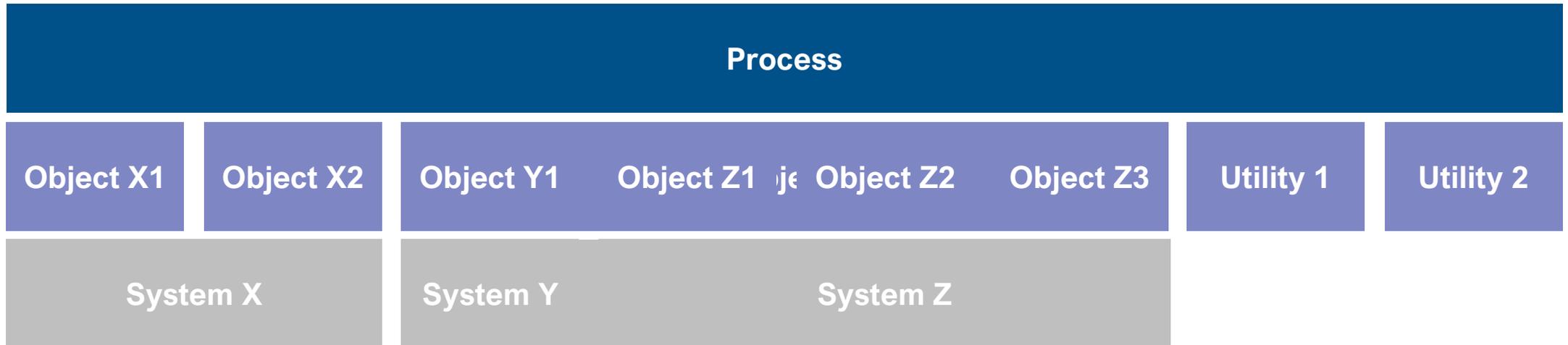
* 로그인 필요



Guide (Guide.pdf)

솔루션 레이어

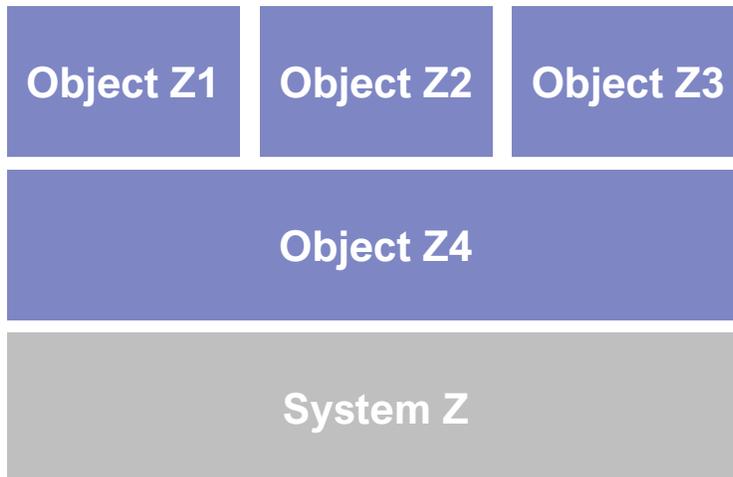
Process Layers





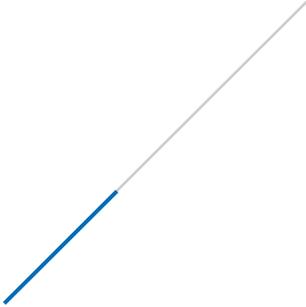
솔루션 레이어

Layered Objects



Layered Processes





3

프로세스 빌딩 & 디버깅

What's

Process Building & Debugging



프로세스 빌딩 & 디버깅

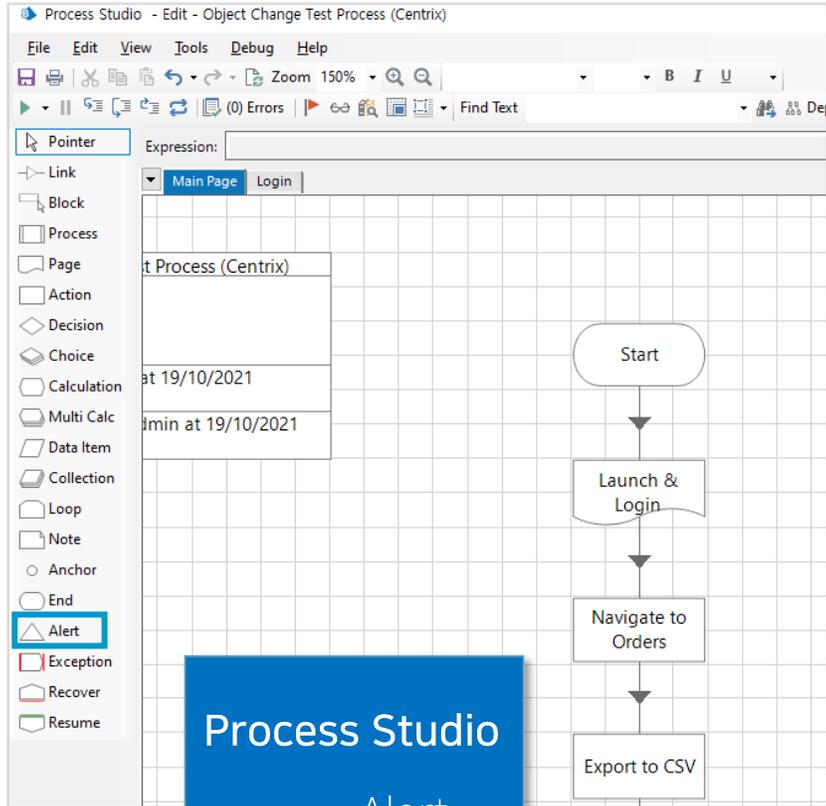
좋은 오브젝트를 만들기 위한 포인트는 가능한 빠른 구동 속도로 많은 케이스를 Control Room에서 테스트 하는 것이다. 디버깅 모드에서 실행하는 것은 Control Room에서 실행 하는 것에 비해 구동 속도가 느리기 때문에 드러나지 않는 문제가 있을 수 있다. 이를 방지하기 위해 반복 프로세스를 만들고 Control Room에서 구동 테스트를 하는 것이 좋다. 이를 기계적 테스트라 하고 이 테스트를 거친 액션은 '기계적으로 강력' 하다 할 수 있다.

데이터를 포함한 테스트 또한 중요하다. 비현실적인 데이터로만 테스트 하는 것은 잠재적 문제를 일으킬 수 있기 때문에, 라이브 환경에서 존재할 수 있는 현실성 높은 데이터로의 테스트가 반드시 필요하다.

이러한 테스트의 목적은, 프로세스 빌드 및 테스트 중 발생할 수 있는 Unknown Exception의 수를 최소화 하는 것이다. 많은 블루프리즘 초심자가 겪는 문제는, 아직 안정화 되지 않은 오브젝트들의 다양한 액션들을 한데 모아 프로세스를 작성한 뒤 테스트를 하여, 버그의 원인파악에 어려움을 겪는 경우가 있다.

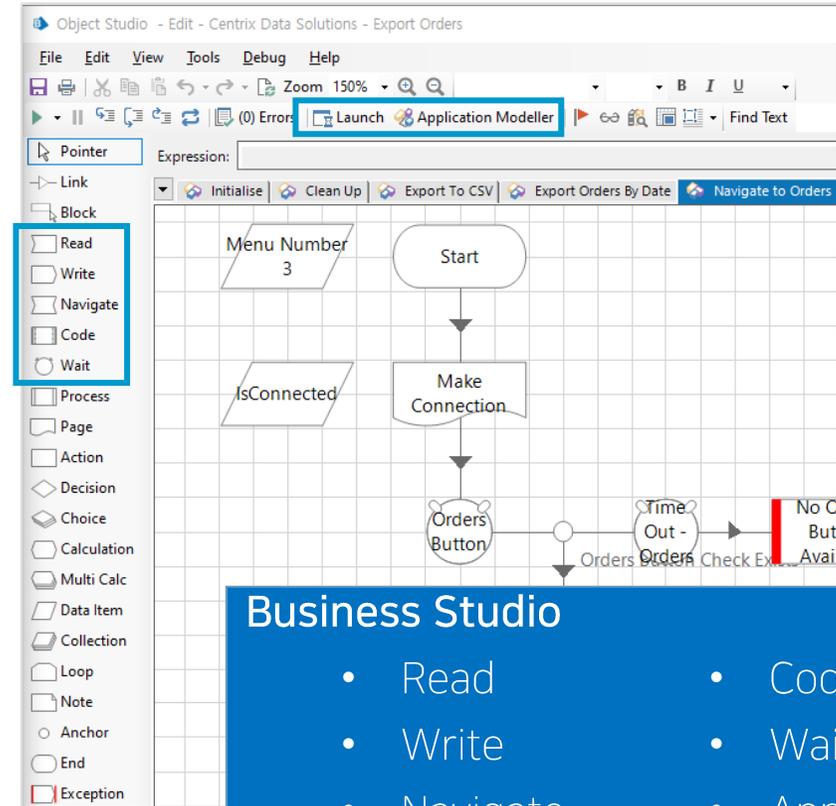
정리하면, 프로세스를 조립할 때 모든 액션들이 이미 견고하다는 확신을 갖을 수 있다면, 프로세스 로직 작성에 집중할 수 있다.

프로세스 스튜디오 vs 오브젝트 스튜디오



Process Studio

- Alert



Business Studio

- Read
- Write
- Navigate
- Code
- Wait
- Application Modeller

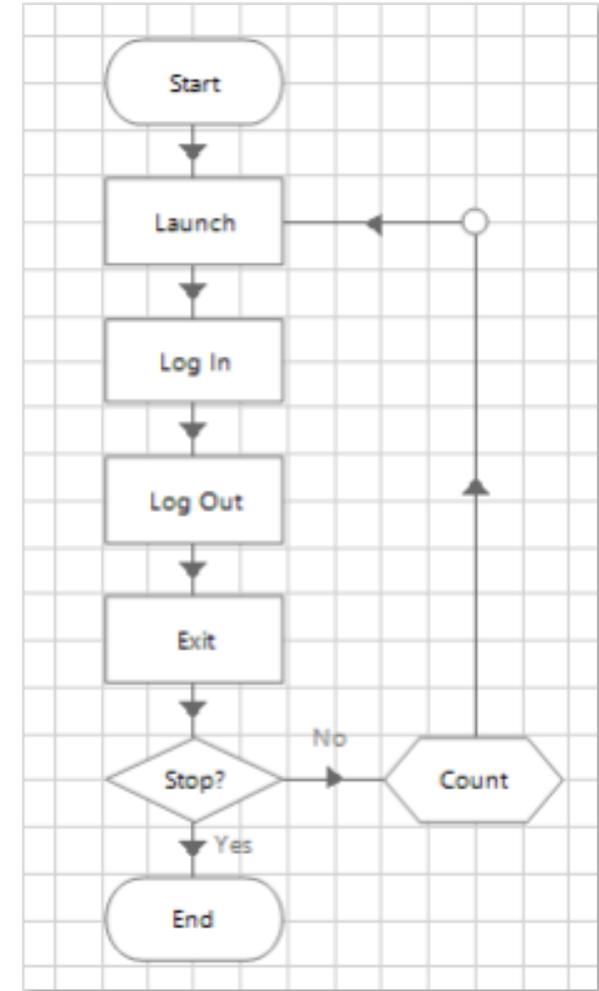
Process Building

기계적 반복 테스트

I 기계적 반복 테스트

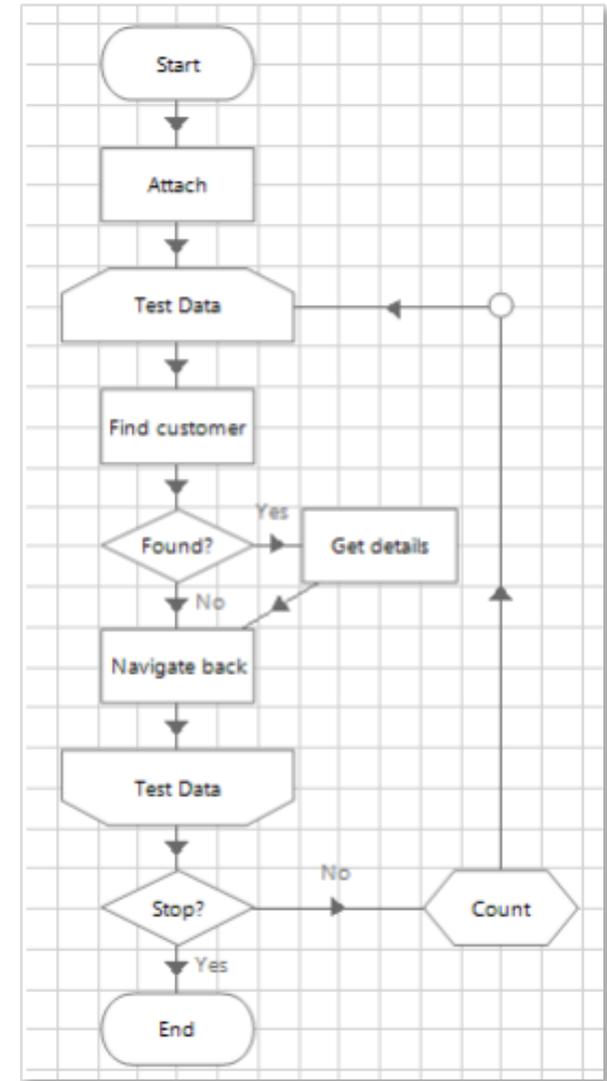
- A. 테스트용 프로세스 제작
- B. 프로세스를 게시(Publish)
- C. Control Room 에서 프로세스를 구동
- D. 문제 발견 시 해당 부분 수정

- 이상이 발생하지 않을 때까지 위 3~4번을 반복.
- 디버그 모드의 최대속도보다 Control Room의 구동속도가 더 빠르다는 것을 인지.



Object Building 데이터와 함께 테스트

- 기본적으로 오브젝트의 로직에는 여러 데이터가 포함
 - 가장 일반적인 실수는 동일한 데이터로만 테스트 하는 것
 - 이는 곧, 테스트 되지 않은 케이스에 대해서는 잘 작동하지 않을 수 있다는 의미
- ❖ 마지막으로, 라이브 데이터와의 테스트도 중요





1

객체지향 오브젝트

블루프리즘에의 Best Practice를 따르는 재사용성이 높은 오브젝트를 통해 불필요한 비용 증가를 최소화 한다.

2

일관된 프로세스

블루프리즘의 프로세스 템플릿 및 워크큐를 활용하여 체계적이고 일관성 있는 프로세스를 제작한다.

3

유지보수의 용이성

객체지향 오브젝트와 일관된 프로세스로 인해 유지보수가 보다 용이하고 비용을 최소화 한다.

* RPA 확산 전, 반드시 이루어야 할 것은 유지보수의 안정화 이다.

4

확산성

품질 높은 오브젝트로 자산화 된 기업의 시스템/어플리케이션들은 프로세스 설계 및 구현 만으로 빠르고 비용효율적으로 확산 된다.

Thank you!

Object & Process 및 Solution Design

세션이 끝났습니다.

잘 따라와 주셔서 감사합니다. 😊

